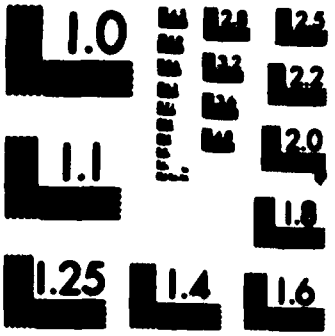
AD-A181 507 FABLE: KNOWLEDGE FOR SEMICONDUCTOR MANUFACTURING(U) 1/1
STANFORD UNIV CA CENTER FOR INTEGRATED SYSTEMS
B DAVIES ET AL. 1986 MDA903-80-C-0432

UNCLASSIFIED F/G 13/8 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

FABLE: KNOWLEDGE FOR SEMICONDUCTOR MANUFACTURING

Byron Davies[1], Steven D. Leeke, and Krishna Saraswat

The Center for Integrated Systems
Stanford University
Stanford, CA 94305

[1]Visiting Scholar from Texas Instruments

1. The Stanford FABLE Project

The Stanford FABLE project in semiconductor manufacturing science is working to change the nature of semiconductor manufacturing. The breadth of modern semiconductor products depends on mass production of DRAMs, to debug and refine new processes before they are utilized for lower volume but higher profit products. The FABLE project is developing methodologies to permit the rapid development and engineering of new processes so that highly innovative products depending on new processes can be fielded more rapidly.

A key goal of the FABLE project is the *programmable factory*, an integrated system of manufacturing equipment, sensors, and computer hardware and software. Just as a programmer can rapidly modify and debug a complex computer program, so a process engineer should be able to modify and debug the complex processes which controls the manufacturing of semiconductors. To make the factory easier to program, a powerful process CAD system is being developed.

A second key goal of the FABLE project is the *virtual factory*, a factory that can be run in simulation. Just as VLSI circuits are simulated before they are cast in silicon, so VLSI manufacturing processes should be simulated before they are run in the factory. We are developing a powerful computer-based simulation system that will permit us to simulate semiconductor manufacturing processes in their entirety, using knowledge about equipment, processes, materials, devices, and circuits to predict critical measurements of manufacturing performance such as yield, electrical performance, throughput, and equipment utilization.

The programmable factory and the virtual factory must be based on a large common knowledge base that captures knowledge about equipment, processes, materials, schedules and other aspects of semiconductor manufacturing. The different software systems needed to support process development tasks -- e.g., design, debugging, execution, data acquisition/interpretation, control, updating -- need to have access to similar knowledge. For efficiency of development and ease of maintenance, it is crucial that information about a piece of equipment, for example, not be encoded one way for process design and another way for process debugging. This argues for a declarative (as structure and statements) rather than procedural (as programs) representation of the knowledge[1, 2]. Specialized interpreters will exploit the common knowledge base for each task.

The focus of the FABLE project is a growing body of knowledge about semiconductor manufacturing together with a set of techniques and tools for encoding and using that knowledge to support the automated factory. The structure of a part of the FABLE knowledge base is shown in Figure 1. The development of the knowledge base and the tools are driven in part by questions such as the following:

What does the programmable factory need to know? Among

other things, the automation system must know about (1) the workings of the factory itself: equipment, equipment operations, material flows; (2) the nature of potential problems so they can be anticipated and prevented, or diagnosed and fixed; and (3) the history of the operation of the factory, to recognize long-term trends.

How can such knowledge be encoded for use by the automated factory? A variety of formalisms are possible, all of which can be reduced to first order logic and its extensions[2]. For initial development, we have chosen a primarily frame-based representation system, KEE[3]. This permits the easy entry and editing of descriptions of objects in a classification hierarchy. The FABLE knowledge base is available to programs running in Common Lisp on a TI Explorer/LX Lisp machine. This Lisp machine provides both Lisp *and* Unix System V environments, allowing access to knowledge-based tools in Lisp and traditional simulation programs on Unix.

How can the knowledge be efficiently acquired and updated? Efficient knowledge acquisition is critical, because semiconductor manufacturing knowledge changes rapidly. We are developing interface tools to permit the manufacturing expert to enter knowledge directly, without having to use an intermediary. For example, we have implemented general tools to facilitate acquisition of related classification hierarchies, and domain specific tools for acquiring and displaying process descriptions.

These questions are stimulated by FABLE-related projects in factory simulation, equipment modeling, intelligent processing equipment automation, automated test structure interpretation, and graphical equipment interfaces. Section 2 describes one such project, confirming the importance of the explicit use of knowledge in manufacturing automation systems.

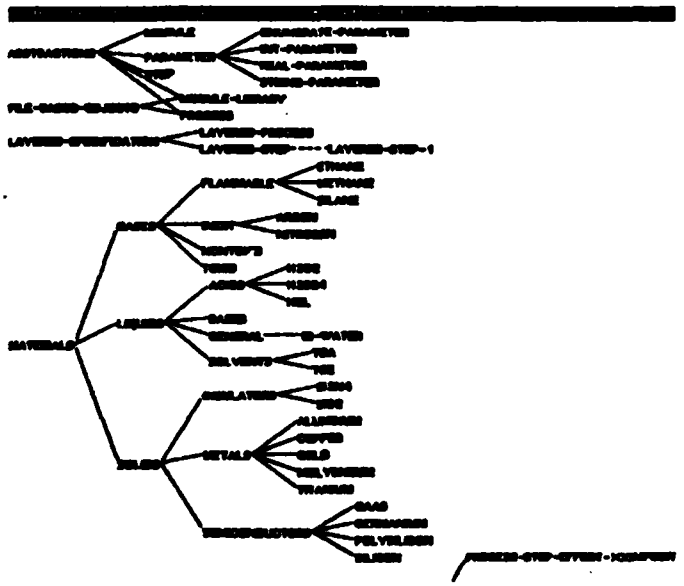


Figure 1: A portion of the FABLE knowledge base

2. Tools for Process Specification and Simulation

The most pervasive and recognizable form of centralized knowledge in a wafer fabrication line is the process runsheet. While this form of process specification embodies only a part of the complete body of knowledge required for wafer processing it nevertheless remains largely isolated from and underutilized by other tools such as simulation programs.

To address the traditional simulation area we have developed a flexible Unix-based system in C that aids process specification and simulation at the level of the SUPREM[4] family of process simulation tools. Our system is called SHIPS, short for Stanford High-Level Incremental Process Simulation. The purpose of this tool is to allow a process engineer to specify and simulate a VLSI process using physical process models so as to facilitate statistically significant simulation results. This requires computationally intensive process simulation and the statistical analysis of results. We refer to the methodology embodied in this tool as *manufacturing-level process specification* and *simulation*. Previous efforts in this area have come from Carnegie-Mellon and Hitachi, with the FABRICS family of programs and the CASTAM program, respectively[5, 6].

SHIPS utilizes incremental simulation, a compact simulation language that is structured according to a set of abstractions that are familiar to processing personnel, and a menu-driven, form-based user interface that provides a productive framework for creating new processes and running simulations. Moreover, the system is designed to perform the simulations in a software simulation *engine* independent manner. That is, the fundamental physical simulator is considered a simulation engine, and not an integral part of the SHIPS system. SHIPS drives the software simulation engine to perform the simulations and analyzes the corresponding output. For our work the SUPREM family of process simulators is used as the software simulation engine.

A knowledge-based version SHIPS in Lisp under KEE is currently under development on the Explorer/LX. This version of SHIPS will play a significant part in knowledge-based process specification and the development of the virtual factory for VLSI manufacturing modeling. A complete process specification will contain both the *physical* information required for specifying the wafer treatments and the expected effects, as well as the *production* information necessary to perform the actual manufacturing process. Figure 2 shows a prototype interface for SHIPS under KEE. This interface is designed to allow the iconic representation and manipulation of processes as collections of modules, which are in turn collections of individual steps. Modules are grouped along logical divisions and placed in libraries. The sequential ordering of the steps and modules is represented with arrows connecting the icons, and rework routes are implemented with conditional branches based on a measured value or the result of an inspection.

The C version of SHIPS is designed to meet the needs of the large body of process simulation users who need to perform simulations to determine the manufacturability of a process. We have seen in our work, however, that the advantages of a knowledge-based system over a traditional one are significant. The flexibility and adaptability of the knowledge embedded in the KEE knowledge-base stands in glaring contrast to the harsh inflexibility of the same knowledge in C. This difference makes updating the KEE version much easier.

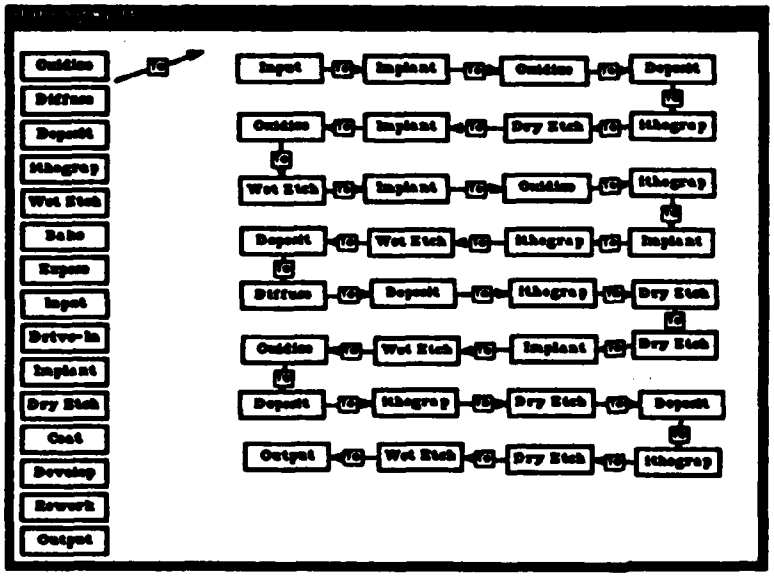


Figure 2: A prototype of the SHIPS interface in KEE

3. Conclusion

The SHIPS example demonstrates the value of the knowledge based approach to building semiconductor manufacturing application systems. The FABLE project will continue the development of knowledge bases and associated tools to support such applications.

4. Acknowledgements

The authors wish to thank Dr. John Shott of the Stanford IC Lab for his insights into processing, and Texas Instruments for its support of this work and the CIS. This work was supported by SRC contract 84-01-046 and DARPA contract MDA 903-80-C-0432.

References

1. Harold L. Ossher and Brian K. Reid, "Fable: A Programming-language Solution to IC Process Automation Problems", Tech. report 248, Stanford University, September 1983.

2. Michael R. Genesereth and Nils J. Nilsson, *Logical Foundations of Artificial Intelligence*, Forthcoming, 1987.

3. IntelliCorp, *Kee Software Development System User's Manual Version 3.0*, 1986.

4. C.P. Ho, J.D. Plummer, S.E. Hansen, and R.W. Dutton, "VLSI Process Modeling - Suprem III", *IEEE Transactions on Electron Devices*,, Vol. ED-30, No. 11, November, 1983, pp. 1438-1453.

5. Yukio Aoki, Toru Toyabe, Shojiro Asai, and Takaaki Hagiwara, "CASTAM: A Process Variation Analysis Simulator for MOS LSI's", *IEEE Transactions on Electron Devices*,, Vol. ED-31, No. 10, October, 1984, pp. 1462-1467.

6. Sani R. Nassif, Andrzej J. Strojwas, and Stephen W. Director, "FABRICS II: A Statistically Based IC Fabrication Process Simulator", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*,, Vol. CAD-3, No. 1, January, 1984, pp. 40-46.

END

7-87

DTIC